NASA-CR-193086

NAG – 1 – 1466

ADJOINT METHODS FOR AERODYNAMIC WING DESIGN

NLPN 92-737

GRANT
IN-05-CR
164775
P.18

Semi-Annual Progress Report

May 1993

Principal Investigator: Bernard Grossman

Department of Aerospace and Ocean Engineering
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061

Progress on NASA Research Grant NAG–1–1466 is summarized in the following draft of a paper which is included with this report:

- Narducci, R., Grossman, B. and Haftka, R. T., "Design Sensitivity Algorithms for an Inverse Design Problem Involving a Shock Wave", paper submitted to the AIAA 32nd Aerospace Sciences Meeting, Jan. 1994.

Design Sensitivity Algorithms for an Inverse Design Problem Involving a Shock Wave

R. Narducci, B. Grossman, R.T. Haftka

Virginia Polytechnic Institute and State University
Dept. Aerospace and Ocean Engineering
Blacksburg, VA 24061

A model inverse design problem is used to investigate the effect of flow discontinuities on the optimization process. The optimization involves finding the cross-sectional area distribution of a duct that produces velocities that closely match a targeted velocity distribution. Quasi-one-dimensional flow theory is used, and the target is chosen to have a shock wave in its distribution. The objective function which quantifies the difference between the targeted and calculated velocity distributions may become non-smooth due to the interaction between the shock and the discretization of the flow field. This paper offers two techniques to resolve the resulting problems for the optimization algorithms. The first, shock-fitting, involves careful integration of the objective function through the shock wave. The second, coordinate straining with shock penalty, uses a coordinate transformation to align the calculated shock with the target and then adds a penalty proportional to the square of the distance between the shocks. The techniques are tested using several popular sensitivity and optimization methods, including finite-differences, and direct and adjoint discrete sensitivity methods Two optimization strategies, Gauss-Newton and sequential quadratic programming (SQP), are used to drive the objective function to a minimum.

INTRODUCTION

Solutions to high speed aerodynamic design problems generally require numerical solutions of the Euler or Navier Stokes equations. These flows often contain regions of steep gradients such as shock waves, contact surfaces, and boundary and shear layers. Analysis of these problems are computationally expensive. In the context of optimized design which places an even greater demand on computational resources, it is important that efficient sensitivity and optimization algorithms are studied. In particular the study focuses on the effect of shock waves in the design optimization process.

Recently, Frank and Shubin (Ref. 1 and 2) studied the simple model problem of inviscid compressible flow through a variable area duct. They formulated an inverse design problem and investigated several design sensitivity techniques. Although their designs contained shock waves, their results did not indicate any adverse effects of shock waves on the design optimization process.

This appears to be due to the fact that their initial conditions always placed the shock at or very near the targeted shock position.

The objective of the present work is to reexamine the model problem considered by Frank and Shubin and to develop efficient strategies for treating flows with shock waves. The model problem studied in Ref. 1 and 2 offers several simplifications which are useful for this study. First, because of the quasi-one-dimensional flow approximation, e.g., Anderson (Ref. 3), an exact algebraic solution may be found, so that discretization errors of the numerical solution can be accurately computed. Second, accurate finite-volume or finite-difference solutions to the governing Euler equations may be found efficiently with shocks captured very accurately using modern computational fluid dynamics techniques.

Following the inverse design problem of Ref. 1 and 2, we attempt to determine a geometry which closely approximates prescribed flow solutions. Inverse problems of this type are useful for this study, since a prescribed flow field can correspond to a known geometry, thereby giving an accurate measure of design errors. Specifically, the design variables are points defining the duct area distribution as a cubic spline. This differs slightly from Ref. 1 and 2, where Frank and Shubin used coefficients of B-splines. In this work B-splines are also implemented to compare with their results.

Our treatment of the shock waves involves the use of the method of strained coordinates for perturbations of transonic flows with shock waves, introduced by Nixon (Ref. 4). This method has been applied for airfoil approximations by Stahara (Ref. 5) and utilized to find sensitivity derivatives in an airfoil optimization by Joh, Grossman, and Haftka (Ref. 6).

This paper first reviews the governing equations for quasi-one-dimensional flow through a variable area duct, boundary conditions and the flow solutions. In the next sections the inverse design problem is formulated and the objective function is examined closely. The non-smooth nature of the objective function for flows with shock waves is shown to lead to convergence difficulties in the optimization problem. Methods of treating this difficulty, including shock-fitting and the coordinate straining are described. Next the optimization procedures used for this problem are discussed. Then finite-difference, direct-discrete and adjoint-discrete methods for computing design sensitivities are presented. Results and a discussion of the advantages and disadvantages of the sensitivity and optimization algorithms conclude this report.

ANALYSIS

Governing Equations

The governing equations for steady, quasi-one-dimensional flow through a duct of varying cross sectional area are the Euler equations,

$$\begin{Bmatrix} \rho uA \\ (\rho u^2 + p)A \\ (\rho e_o + p)uA \end{Bmatrix}_x + \begin{Bmatrix} 0 \\ -pA_x \\ 0 \end{Bmatrix} = 0. \qquad (1)$$

The domain, x, varies from 0 to 1, ρ is the density, u is the velocity, A is the area, p is the pressure, and e_o is the total energy per unit mass. The equation of state for a perfect gas closes the system,

$$p = (\gamma - 1)\rho e, \qquad (2)$$

where e is the energy per unit mass, and γ is the ratio of specific heats assumed to be constant. Following the derivation in Ref. 1, the first and third components of (1) may be integrated directly, and after algebraic manipulation the system is reduced to a single ordinary differential equation in u,

$$f_x + g = 0, \qquad (3)$$

where

$$f(u) = u + \frac{(\gamma - 1)}{(\gamma + 1)} \frac{2h_o}{u}, \qquad (4)$$

$$g(u) = \frac{A_x}{A} \frac{\gamma - 1}{\gamma + 1} \left(u - \frac{2h_o}{u}\right), \qquad (5)$$

and h_o is the total enthalpy per unit mass.

We specify inlet and exit velocities to get a unique solution to the governing differential equations. Velocity boundary values are normalized with respect to the speed of sound at the inlet and are 0.506 and 1.299 for inlet and exit conditions, respectively. These conditions were used in Ref. 1.

The Euler equations are solved using three methods to aid in understanding the effects of the sharpness of the shock on the design process. One method is analytic and an exact solution is obtained. Two finite-volume formulations are used to get approximate solutions. The finite-volume methods used are Godunov and artificial viscosity.

Exact Solution

The exact solution is arrived at by integrating (3) over regions where the solution is smooth. The result from Ref. 1 is

$$Au(2h_o - u^2)^{1/(\gamma - 1)} = k, \qquad (6)$$

where k is the constant of integration. Equation (6) is applied over the region from the inlet to the position of the shock wave, and again from the position of the shock wave to the exit. The constant, k, increases across the shock and is found from the boundary conditions at the inlet and exit for each application of the solution. The position of the shock which is the boundary of the right and left solution is determined by satisfying the Rankine-Hugoniot relation

$$u_L u_R = u_*^2 = 2h_o \frac{\gamma - 1}{\gamma + 1}, \qquad (7)$$

where u_L and u_R are the left and right values of the velocity at the shock. The method for obtaining the exact solution is shown graphically in figure 1.

Finite-Volume Solutions

A numerical solution to (3) is obtained by adding an unphysical time derivative and marching to a steady state,

$$u_t + f_x + g = 0. \tag{8}$$

Time integration is performed using Jameson four-stage Runge-Kutta. Distance along the duct is discretized with u_j evaluated at the cell centers and fluxes evaluated at the cell faces. The spatial derivative is replaced with the finite-volume formulation

$$u_t + \frac{f_{j+1/2} - f_{j-1/2}}{\Delta x} + g_j = 0. \tag{9}$$

Using the Godunov scheme described in Ref. 1, the fluxes are computed according to

$$f_{j+1/2} = \begin{cases} f_{j+1}, & u_j, u_{j+1} < u_* \\ f_j, & u_j, u_{j+1} > u_* \\ f_*, & u_j < u_* < u_{j+1} \\ \max(f_j, f_{j+1}), & u_{j+1} < u_* < u_j \end{cases}, \tag{10}$$

where $f_{j+1} = f(u_{j+1})$, etc., and * indicates sonic flow. This formulation yields a solution containing a sharp shock. An alternative to the Godunov formulation is the artificial viscosity scheme used in Ref. 1 as

$$f_{j+1/2} = \frac{1}{2}\left[f_{j+1} + f_j - \alpha(u_{j+1} - u_j)\right], \tag{11}$$

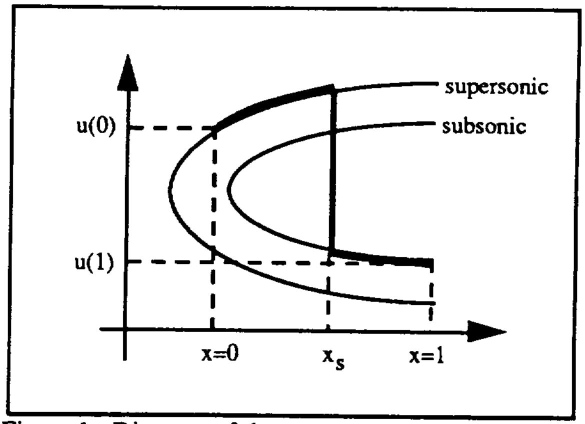

Figure 1: Diagram of the exact solution algorithm as described in Ref. 1.

where α is an artificial viscosity parameter which is related to the numerical dissipation.

The exact solution resolves the shock precisely, while the Godunov scheme smears the shock over two grid points. The artificial viscosity parameter is a control of the amount of shock smearing. For example, $\alpha = 1$ and $\alpha = 4$ spread the shock over approximately 5% and 10% of the domain respectively; $\alpha = 12$ virtually eliminates the discontinuity. The exact and numerical solutions for the target area distribution are compared in figure 2 using a computational domain of 41 grid points.

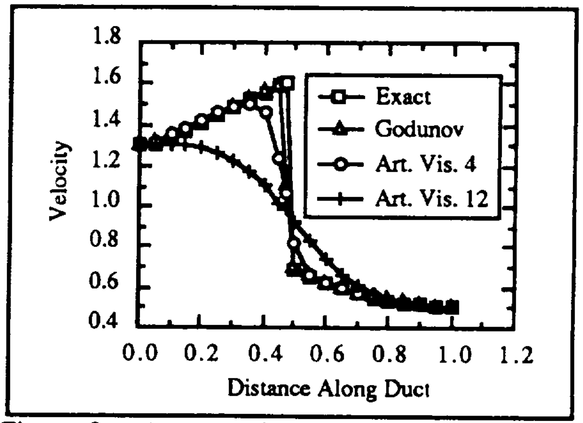

Figure 2: A comparison of solutions containing shocks.

DESIGN PROBLEM

Area Description

The design problem involves finding an area distribution so that the solution to (3) closely matches a given velocity distribution containing a shock. The area distribution is computed at discrete points in the domain from a set of design variables, ξ. Thus the solution to the design problem is expressed as a set of design variables.

The formulation of the area distribution from the design variables is not unique. In this work, a cubic spline is fitted through the area at specified points along the duct as shown schematically in figure 3. The design variables to be optimized are the values of the area at these points. Inlet and exit areas are fixed at normalized values of 1.05 and 1.745. Clamped boundary conditions, i.e. zero slope at the ends are imposed to determine the spline uniquely. Design cases presented in this paper contain up to 20 design variables evenly distributed along the interior of the duct.

An alternate method of formulating the area from design variables involves B-splines. This formulation e.g., Gerald and Wheatly Ref. 7, provides smooth curves and is implemented in this study for the purpose of comparing to the work of Frank and Shubin.

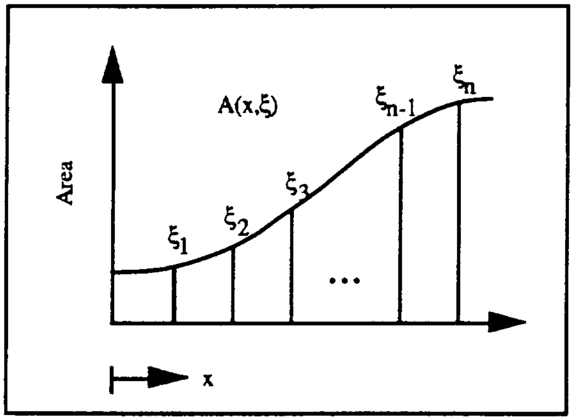

Figure 3: Design variables describing the area distribution

A target velocity distribution was created by solving (3) using an area distribution described by the cubic

$$A(x) = -1.39x^3 + 2.085x^2 + 1.05. \quad (12)$$

This area profile has the properties of A(0) = 1.05, A(1) = 1.745, and zero slope at the ends. The cubic spline formulation can match (12) exactly and will serve as a check on our final design.

A Non-smooth Objective Function

To quantify how well a calculated velocity distribution compares to the target, we define the objective function

$$I(\xi) = \frac{1}{2}\int_0^1 (\hat{u} - u)^2 dx, \quad (13)$$

where $\hat{u} = \hat{u}(x)$ is the target velocity distribution through the duct, and $u = u(x;\xi)$ is the calculated velocity distribution. The velocities are normalized by the speed of sound at the inlet. In the discretization of the problem, the integral is approximated using the trapezoidal rule

$$I(\xi) \cong \frac{1}{2}\Big[\frac{1}{2}(r_1^2 + r_N^2) + \sum_{i=2}^{N-1} r_i^2\Big], \qquad (14)$$

where

$$r = (\hat{u} - u)\sqrt{\Delta x}\,, \qquad (15)$$

and N is the number of grid points. Boundary conditions specify u at the inlet and exit to match the target exactly reducing (14) to

$$I(\xi) \cong \frac{1}{2}\sum_{i=2}^{N-1} r_i^2. \qquad (16)$$

An optimum design is achieved when (16) is a minimum.

Making the trapezoidal rule approximation without regard to the position of the shock wave results in a non-smooth objective function. In figure 4, the objective function is drawn for a design problem involving one design variable and exact solutions to the governing equations of fluid motion. In this case the objective function is discontinuous.

The jumps in the objective function result from the combination of the shock wave and the numerical evaluation of the objective function. The calculation of the objective function (16) does not involve any specific information about the location of the shock. Thus for small perturbations of the area distribution, provided the shock remains between the same grid points, the value of the objective function changes very little. The objective function is dominated by the differences in the target and calculated velocity in the segment between the shocks. Figure 5 is a typical plot of $r = (\hat{u} - u)\sqrt{\Delta x}$ and demonstrates this effect. For perturbations of the area distribution which just moves the shock across the grid line, the objective function changes dramatically. On the other hand, a perturbation which moves the shock just inside the grid line will have almost no effect.

Without a clear picture of the nature of the objective function, one might try increasing the number of grid points to reduce round-off errors involved in the calculation of the design sensitivities. However, this will only create more "stairs" in the objective function

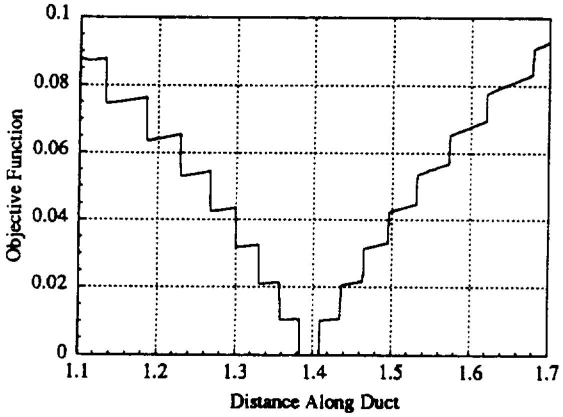

Figure 4: Objective function plot for a one design variable case using an exact flow solver.

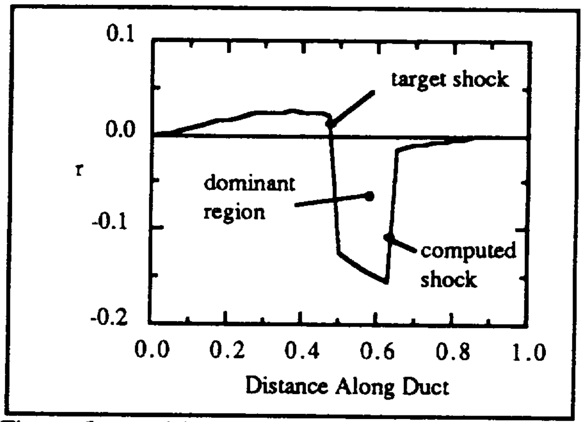

Figure 5: r and hence the objective function dominated by the region between the shock waves.

as now there are more cell centers for which the shock wave passes through. Thus instead of helping, the problem is worsened.

Using a numerical solution for the flow solver, the resolution of the shock decreases, and the objective function while no longer discontinuous, remains non-smooth. Figure 6 is a plot of the objective function calculated via the Godunov scheme, which smears the shock over 2 grid points. For a highly smeared shock, computed using the artificial viscosity scheme with $\alpha = 1$, the objective function appears smooth (figure 7). The smooth objective function has a clear advantage over the discontinuous one, unfortunately this comes at the expense of the accuracy of the flow solution. However, as the grid is refined, the problem will reappear.

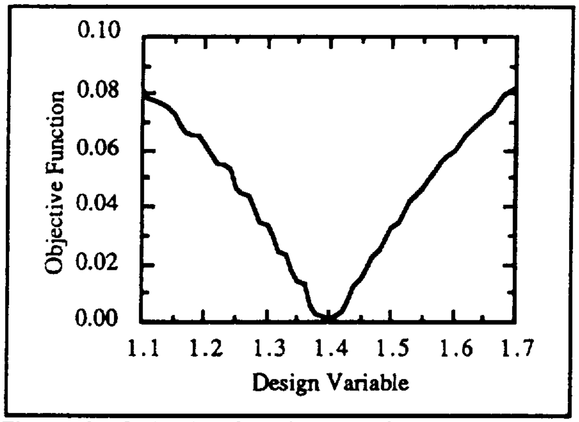

Figure 6: Objective function plot for a one variable design problem using Godunov flow solver.

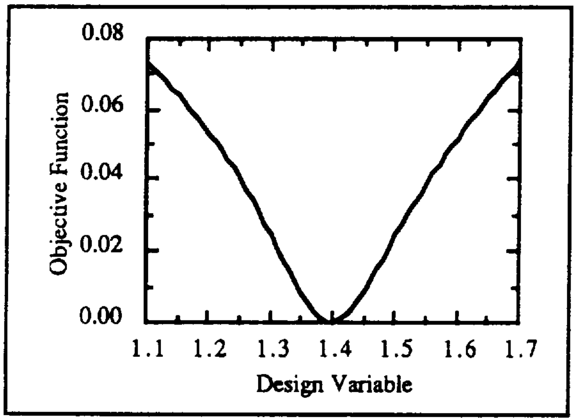

Figure 7: Objective function plot for a one variable design problem using an artificial viscosity solver.

Shock-Fitting

A more precise evaluation of the integral in (13) involves first dividing the integral at the location of the discontinuities and then applying the trapezoidal rule to each segment of the function. The objective function contains two shocks, one from the target distribution and one from the calculated distribution, thus the integral is divided into three segments,

$$I = \int_0^{\hat{x}_s} (\hat{u} - u)^2 dx + \int_{\hat{x}_s}^{x_s} (\hat{u} - u)^2 dx + \int_{x_s}^{1} (\hat{u} - u)^2 dx, \qquad (17)$$

where $\hat{x}_s$ and x_s are the positions of the target and calculated shock waves respectively. Implied in this procedure is the knowledge of the precise locations of the shocks and the values of $\hat{u}$ and u on either side of both shocks.

Applying numerical integration in this manner with exact solutions to (3) produces a well-behaved objective function (figure 8). (The small wiggles are due to plotting resolution and will be corrected in the final version of the paper.)

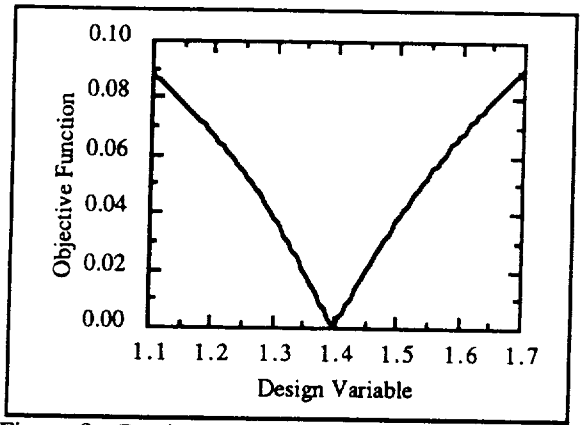

Figure 8: Precise numerical integration of equation 13 yields a smooth objective function

The difficulty in this method arises when the shock is smeared and the jump in velocity across the shock is not obvious. Using a fabricated definition for shock location may locate the shock in a consistent manner from test case to test case, but defining a right and left value for the velocity on either side of the shock is difficult.

Coordinate Straining and Shock Penalty

Another approach to dealing with an objective function which is non-smooth due to the presence of shock waves is the method of coordinate straining as developed by Nixon, Ref. 4. We utilize this method to align the target and the calculated shock waves, effectively ensuring the continuity of the objective function.

The implementation of coordinate straining involves defining a function, s(x), which equals zero at the inlet and exit, and has a value of 1 at the position of the target shock. The function is not unique, and here we choose from Ref. 4,

$$s(x) = \left(\frac{x}{\hat{x}_s}\right)\left(\frac{1-x}{1-\hat{x}_s}\right). \qquad (18)$$

The distance between shocks is defined as

$$\Delta x_s = \hat{x}_s - x_s. \qquad (19)$$

The calculated velocity distribution is strained proportionately to the distance between shocks according to

$$\tilde{u} = u(x - s\Delta x_s). \qquad (20)$$

To apply (20) to in a discrete sense, we determine a grid index M such that $x_M < x_i - s\Delta x_s < x_{M+1}$. Then using linear interpolation, the strained velocity (20) becomes

$$\tilde{u}_i = u_M + \frac{u_{M+1} - u_M}{\Delta x}(x_i - s\Delta x_s - x_M). \qquad (21)$$

Coordinate straining will transform the velocity distribution of figure 9 to that in figure 10.

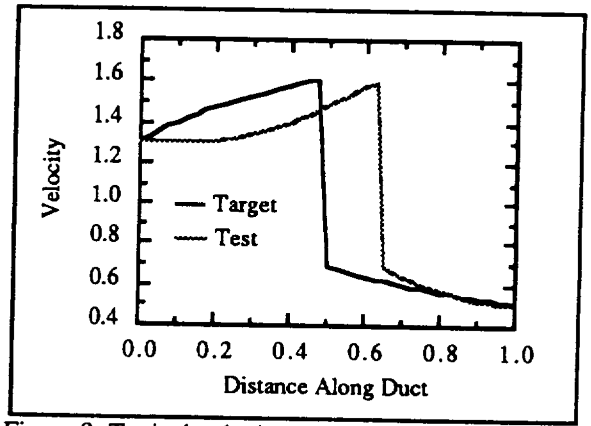

Figure 9: Typical velocity profile of target and calculated velocity distribution. Region between shocks dominates evaluation of the objective function.

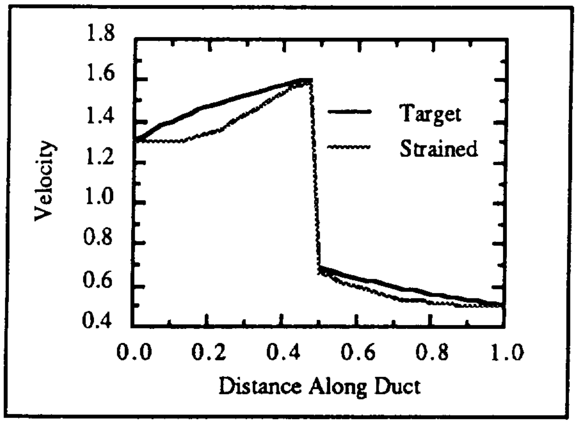

Figure 10: Strained velocity distribution.

In the evaluation of the objective function, using ũ in place of u, the dominating terms which exist in the region between the calculated and target shocks are removed. In the process of removing the staircase, the objective function becomes very flat near the minimum, thus slowing convergence (figure 11). Using this technique, we rely completely on the small differences in velocities outside the shock region to drive the area to the target. Often this is enough to improve the design, but not enough to achieve the best possible one.

To shape the objective function to capture the valley of figure (8), a shock penalty proportional to the square of the difference between the calculated and target shock wave is added to the strained objective function, yielding,

$$I \equiv \frac{1}{2}\left[\sum_{i=2}^{N-1}\tilde{r}_i^2 + \sigma(\hat{x}_s - x_s)^2\right], \qquad (22)$$

where $\tilde{r}_i = (\hat{u}_i - \tilde{u}_i)\sqrt{\Delta x}$, and σ is a positive constant. Values of σ can be chosen so that in the first design iteration (22) equals (13). Figure 12 shows the objective function modified by coordinate straining and shock penalty.

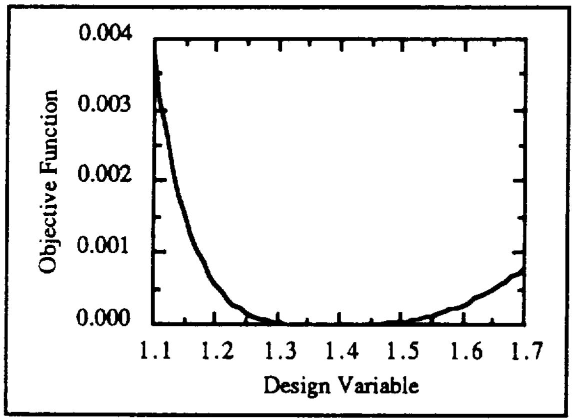

Figure 11: Strained objective function.

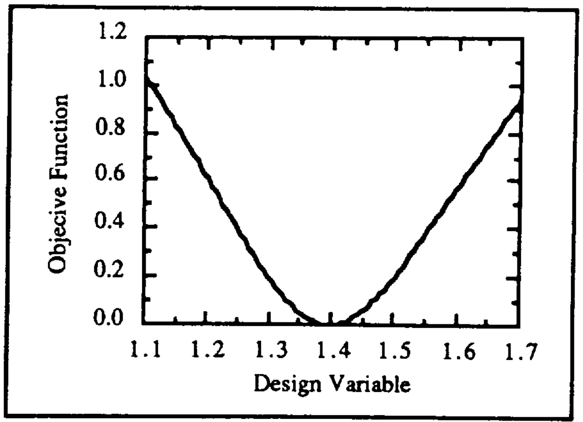

Figure 12: Modified objective function, σ= 5.

OPTIMIZATION ALGORITHMS

Two optimization strategies were used to solve the design problem. One is the Gauss-Newton method which requires some derivation for application to the objective function defined by the coordinate straining and shock penalty technique. The second, an SQP algorithm, can be applied without any special treatment to both the objective functions.

Gauss-Newton Method

In this section the Gauss-Newton optimization algorithm is applied to the function defined by (22). A necessary condition for a minimum is

$$\frac{\partial I}{\partial \xi_j} = \sum_{i=2}^{N-1} \tilde{r}_i \frac{\partial \tilde{r}_i}{\partial \xi_j} - \sigma(\hat{x}_s - x_s)\frac{\partial x_s}{\partial \xi_j} = 0, \quad (23)$$

for j = 1,.., n where n is the number of design variables. The solution to the optimization problem requires finding the root that satisfies $\nabla I(\xi) = 0$. Given the solution at the ℓth iteration, ξ^{ℓ}, we can find $\xi^{\ell+1} = \xi^{\ell} + \Delta\xi^{\ell}$ by expanding $\nabla I(\xi^{\ell+1})$ in a Taylor series and retaining only the first term. Thus using (23) we proceed via a Newton method to get

$$\sum_{i=2}^{N-1} \tilde{r}_i \frac{\partial \tilde{r}_i}{\partial \xi_j} - \sigma(\hat{x}_s - x_s)\frac{\partial x_s}{\partial \xi_j} +$$
$$\sum_{k=1}^{n} \{ \sum_{i=2}^{N-1} [\frac{\partial \tilde{r}_i}{\partial \xi_k}\frac{\partial \tilde{r}_i}{\partial \xi_j} + r_i \frac{\partial^2 r_i}{\partial \xi_k \partial \xi_j}] +$$
$$\sigma \frac{\partial x_s}{\partial \xi_k}\frac{\partial x_s}{\partial \xi_j} - \sigma(\hat{x}_s - x_s)\frac{\partial^2 x_s}{\partial \xi_k \partial \xi_j} \} \Delta\xi$$
$$= 0. \quad (24)$$

Applying (24) to each of n design variables results in a linear system of n equations which drives the design variables to their optimal value. Near the minimum, $\tilde{r}_i$ and $(\hat{x}_s - x_s)$ are small, and the second derivative terms may be neglected. This avoids the computation of the second derivatives and comprises the Gauss-Newton method. The system is then

$$\sum_{k=1}^{n} \Delta\xi_k [\sum_{i=2}^{N-1} \frac{\partial \tilde{r}_i}{\partial \xi_k}\frac{\partial \tilde{r}_i}{\partial \xi_j} + \sigma \frac{\partial x_s}{\partial \xi_k}\frac{\partial x_s}{\partial \xi_j}] =$$
$$- \sum_{i=2}^{N-1} \tilde{r}_i \frac{\partial \tilde{r}_i}{\partial \xi_j} + \sigma \frac{\partial x_s}{\partial \xi_j}. \quad (25)$$

To implement (25) we must accurately compute $\partial r_i/\partial \xi_j$. In the following sections we investigate several methods to compute these derivatives.

SQP Method

The sequential quadratic program method is described in detail in many references, e.g. Haftka and Gürdal, Ref. 8, and no details are provided here. The algorithm used in this work is developed by Schittkowski, Ref 9.

SENSITIVITY METHODS

Finite-Differences

The optimization routines require either $\partial r_i/\partial \xi_j$ and $\partial x_s/\partial \xi_j$ or $\partial I/\partial \xi_j$ which can be computed using finite-differences. In this work, first order forward and backward approximations,

$$\frac{\partial r_i}{\partial \xi_j} = \frac{1}{\Delta\xi}[r_i(\xi_1, \xi_2, \cdots, \xi_j + \Delta\xi_j, \cdots, \xi_n)$$
$$- r_i(\xi_1, \cdots \xi_n)], \quad (26)$$

$$\frac{\partial r_i}{\partial \xi_j} = \frac{1}{\Delta\xi}[r_i(\xi_1, \cdots \xi_n)$$
$$- r_i(\xi_1, \xi_2, \cdots, \xi_j - \Delta\xi_j, \cdots, \xi_n), \quad (27)$$

and second order central difference approximations,

$$\frac{\partial r_i}{\partial \xi_j} = \frac{1}{2\Delta\xi}\Big[r_i(\xi_1,\xi_2,\cdots,\xi_j + \Delta\xi_j,\cdots,\xi_n) \\ - r_i(\xi_1,\xi_2,\cdots,\xi_j - \Delta\xi_j,\cdots,\xi_n), \qquad (28)$$

were investigated. Numerical results did not indicate a clear advantage to any of the three. The central difference approximation requires 2N additional forward solutions, whereas the one sided differences require only N additional forward solutions. Results presented in this paper use the forward difference approximation.

We concluded from a study that a computational domain containing at least 41 grid points and $\Delta\xi = 10^{-4}$ are sufficient for computing the design sensitivities accurately.

Direct Discrete

This method computes the design sensitivities by applying the chain rule of differentiation to the discrete governing equations. While this method is cheaper than the finite-difference approach, it is more involved to implement. Further, the calculation of the sensitivities requires knowledge of the flow solving algorithm.

In this formulation, we distinguish between the flow variables, **u**, and the design variables ξ. The flow variables are the values of the velocity at the grid points in the domain, and are themselves functions of the design variables. In addition to **u**, we include the shock position, x_s, which is also dependent on ξ. Considering the strained and shock penalized objective function we have,

$$I = I[\tilde{\mathbf{u}}(\xi), x_s(\xi)]. \qquad (29)$$

Applying the chain rule to (29) to compute the sensitivity, we have

$$\frac{\partial I}{\partial \xi_j} = \frac{\partial I}{\partial \tilde{\mathbf{u}}}\frac{\partial \tilde{\mathbf{u}}}{\partial \xi_j} + \frac{\partial I}{\partial x_s}\frac{\partial x_s}{\partial \xi_j}, \qquad (30)$$

where

$$\frac{\partial I}{\partial \tilde{\mathbf{u}}} = \left[\frac{\partial I}{\partial \tilde{u}_1} \quad \frac{\partial I}{\partial \tilde{u}_2} \quad \cdots \quad \frac{\partial I}{\partial \tilde{u}_N}\right], \qquad (31)$$

and

$$\frac{\partial \tilde{\mathbf{u}}}{\partial \xi_j} = \left[\frac{\partial \tilde{u}_1}{\partial \xi_j} \quad \frac{\partial \tilde{u}_2}{\partial \xi_j} \quad \cdots \quad \frac{\partial \tilde{u}_N}{\partial \xi_j}\right]^T. \qquad (32)$$

The straining function is defined in (21) and in general is a function of **u** and x_s,

$$\tilde{\mathbf{u}} = \tilde{\mathbf{u}}[\mathbf{u}(\xi), x_s(\xi)]. \qquad (33)$$

Differentiation with respect to the jth design variable yields

$$\frac{\partial \tilde{\mathbf{u}}}{\partial \xi_j} = \frac{\partial \tilde{\mathbf{u}}}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial \xi_j} + \frac{\partial \tilde{\mathbf{u}}}{\partial x_s}\frac{\partial x_s}{\partial \xi_j}, \qquad (34)$$

where

$$\frac{\partial \tilde{\mathbf{u}}}{\partial \mathbf{u}} = \begin{bmatrix} \frac{\partial \tilde{u}_1}{\partial u_1} & \frac{\partial \tilde{u}_1}{\partial u_2} & \cdots & \frac{\partial \tilde{u}_1}{\partial u_N} \\ \frac{\partial \tilde{u}_2}{\partial u_1} & \frac{\partial \tilde{u}_2}{\partial u_2} & \cdots & \frac{\partial \tilde{u}_2}{\partial u_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \tilde{u}_N}{\partial u_1} & \frac{\partial \tilde{u}_N}{\partial u_2} & \cdots & \frac{\partial \tilde{u}_N}{\partial u_N} \end{bmatrix}, \qquad (35)$$

$$\frac{\partial \mathbf{u}}{\partial \xi_j} = \begin{bmatrix} \frac{\partial u_1}{\partial \xi_j} & \frac{\partial u_2}{\partial \xi_j} & \cdots & \frac{\partial u_N}{\partial \xi_j} \end{bmatrix}^T, \qquad (36)$$

and

$$\frac{\partial \tilde{\mathbf{u}}}{\partial x_s} = \begin{bmatrix} \frac{\partial \tilde{u}_1}{\partial x_s} & \frac{\partial \tilde{u}_2}{\partial x_s} & \cdots & \frac{\partial \tilde{u}_N}{\partial x_s} \end{bmatrix}^T. \qquad (37)$$

The shock position is defined where the Rankine-Hugoniot relation (7) is satisfied. In a numerical solution where u_L and u_R are not clearly defined, we take the position of the shock to be the point in a steep compression where the Mach number is one, which may be written as

$$u(x_s;\xi) = u_* = \sqrt{2h_o \frac{\gamma - 1}{\gamma + 1}}. \qquad (38)$$

Discretized, we locate the shock by linear interpolation

$$u_s = u_{js} + \frac{u_{js+1} - u_{js}}{\Delta x}(x_s - x_{js}) = u_*, \ (39)$$

where the shock lies between the js and js+1 grid points. Solving for the shock position yields

$$x_s[\mathbf{u}(\xi)] = x_{js} + \frac{\Delta x}{u_{js+1} - u_{js}}(u_* - u_{js}). \qquad (40)$$

The derivative of the shock position with respect to the jth design variable is thus

$$\frac{\partial x_s}{\partial \xi_j} = \frac{\partial x_s}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \xi_j}, \qquad (41)$$

where

$$\frac{\partial x_s}{\partial \mathbf{u}} = \begin{bmatrix} \frac{\partial x_s}{\partial u_1} & \frac{\partial x_s}{\partial u_2} & \cdots & \frac{\partial x_s}{\partial u_N} \end{bmatrix}. \qquad (42)$$

Substituting (34) and (41) into (30) gives us an expression for the design sensitivities

$$\frac{\partial I}{\partial \xi_j} = \mathbf{V}^T \frac{\partial \mathbf{u}}{\partial \xi_j} \qquad (43)$$

where

$$\mathbf{v}^T \equiv \frac{\partial I}{\partial \tilde{\mathbf{u}}}\left(\frac{\partial \tilde{\mathbf{u}}}{\partial \mathbf{u}} + \frac{\partial \tilde{\mathbf{u}}}{\partial x_s}\frac{\partial x_s}{\partial \mathbf{u}}\right) + \frac{\partial I}{\partial x_s}\frac{\partial x_s}{\partial \mathbf{u}}. \ (44)$$

From equations (21), (22), and (40) we can obtain analytic expression to evaluate all the derivatives in (43) with the exception of $\partial \mathbf{u}/\partial \xi_j$. The direct discrete method applies the chain rule to the governing equations discretized by some numerical scheme. We have N discretized equations that are, in general, functions of the flow variables and the design variables. These equations are given the symbol w,

$$\begin{aligned}
w_1 &= w_1(u_1,\ldots,u_N,\xi_1,\ldots\xi_n) = 0\\
w_2 &= w_2(u_1,\ldots,u_N,\xi_1,\ldots\xi_n) = 0\\
&\vdots\\
w_N &= w_N(u_1,\ldots,u_N,\xi_1,\ldots\xi_n) = 0. \qquad (45)
\end{aligned}$$

Differentiating (45) with respect to the jth design variable, we find

$$\frac{\partial \mathbf{w}}{\partial \xi_j} = \left(\frac{\partial \mathbf{w}}{\partial \xi_j}\right)_{\mathbf{u}} + \mathbf{J}\frac{\partial \mathbf{u}}{\partial \xi_j} = 0, \qquad (46)$$

where

$$\mathbf{w} = \begin{bmatrix} w_1 & w_2 & \ldots & w_N \end{bmatrix}^T, \qquad (47)$$

$$\frac{\partial \mathbf{w}}{\partial \xi_j} = \begin{bmatrix} \frac{\partial w_1}{\partial \xi_j} & \frac{\partial w_2}{\partial \xi_j} & \cdots & \frac{\partial w_N}{\partial \xi_j} \end{bmatrix}^T, \qquad (48)$$

and

$$\mathbf{J} \equiv \left(\frac{\partial \mathbf{w}}{\partial \mathbf{u}}\right)_{\xi} = \begin{bmatrix} \frac{\partial w_1}{\partial u_1} & \frac{\partial w_1}{\partial u_2} & \cdots & \frac{\partial w_1}{\partial u_N} \\ \frac{\partial w_2}{\partial u_1} & \frac{\partial w_2}{\partial u_2} & \cdots & \frac{\partial w_N}{\partial u_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial w_N}{\partial u_1} & \frac{\partial w_N}{\partial u_2} & \cdots & \frac{\partial w_N}{\partial u_N} \end{bmatrix}. \qquad (49)$$

We can find $\partial \mathbf{u}/\partial \xi_j$ by solving the linear system

$$\mathbf{J}\frac{\partial \mathbf{u}}{\partial \xi_j} = -\left(\frac{\partial \mathbf{w}}{\partial \xi_j}\right)_{\mathbf{u}}. \qquad (50)$$

Equation (50) must be solved n times to find the sensitivities with respect to all the design variables. For other problems where N is extremely large, it may not be possible to store the L-U decomposition of **J**. In such a case solving (50) n times would be expensive.

Adjoint Discrete

A third method to compute design sensitivities from the discrete equations is the adjoint method. We define an augmented objective function

$$I^* = I + \lambda^T \mathbf{w}, \qquad (51)$$

where I is defined in (22), λ^T is a row vector of Lagrange multipliers and **w** is the column vector of discretized governing equations. The sensitivity with respect to the jth design variable is

$$\frac{\partial I^*}{\partial \xi_j} = \frac{\partial I}{\partial \xi_j} + \lambda^T \frac{\partial \mathbf{w}}{\partial \xi_j}. \qquad (52)$$

The first term on the right hand side has been expanded in the derivation of the direct discrete method (equation (43)). The derivative of the governing equations has also been expanded in equation (46). Substituting (43) and (46) into (52) yields

$$\frac{\partial I^*}{\partial \xi_j} = \mathbf{v}^T \frac{\partial \mathbf{u}}{\partial \xi_j} + \lambda^T\left[\left(\frac{\partial \mathbf{w}}{\partial \xi_j}\right)_{\mathbf{u}} + \mathbf{J}\frac{\partial \mathbf{u}}{\partial \xi_j}\right]. \qquad (53)$$

We rearrange (53) to collect terms multiplying $\partial \mathbf{u}/\partial \xi_j$ to get

$$\frac{\partial I^*}{\partial \xi_j} = \left(\mathbf{v}^T + \lambda^T \mathbf{J}\right)\frac{\partial \mathbf{u}}{\partial \xi_j} + \lambda^T\left(\frac{\partial \mathbf{w}}{\partial \xi_j}\right)_u . \qquad (54)$$

The vector of Lagrange multipliers is arbitrary. If λ is chosen such that

$$\mathbf{J}^T\lambda = -\mathbf{v}, \qquad (55)$$

then the sensitivity is

$$\frac{\partial I}{\partial \xi_j} = \frac{\partial I^*}{\partial \xi_j} = \lambda^T\left\{\frac{\partial \mathbf{w}}{\partial \xi_j}\right\}_u . \qquad (56)$$

The advantage this method has over the direct discrete method is evident with very large systems. Equation (55) has to be solved only once to calculate all the Lagrange multipliers. The sensitivities maybe then computed inexpensively using (56).

Comparison

For a design case involving three design variables and the artificial viscosity flow solver, sensitivities have been recorded in tables 1, 2, and 3. Each table contains sensitivities calculated by each method. Table 1 corresponds to the discontinuous objective function, table 2 uses coordinate straining, and table 3 uses coordinate straining with a shock penalty. We see that all three methods produce similar results for the design sensitivities.

	Finite-Difference	Direct Discrete	Adjoint Discrete
$\partial I/\partial \xi_1$	-4.914e-2	-4.921e-2	-4.921e-2
$\partial I/\partial \xi_2$	6.200e-2	6.231e-2	6.231e-2
$\partial I/\partial \xi_3$	-2.995e-1	-3.003e-1	-3.003e-1

Table 1: Sensitivity comparison using the discontinuous objective function.

	Finite-Difference	Direct Discrete	Adjoint Discrete
$\partial I/\partial \xi_1$	2.312e-3	2.305e-3	2.305e-3
$\partial I/\partial \xi_2$	4.012e-4	3.996e-4	3.996e-4
$\partial I/\partial \xi_3$	-2.514e-3	-2.534e-3	-2.534e-3

Table 2: Sensitivity comparison using the coordinate strained objective function.

	Finite-Difference	Direct Discrete	Adjoint Discrete
$\partial I/\partial \xi_1$	-1.355e-2	-1.369e-2	-1.369e-2
$\partial I/\partial \xi_2$	-5.326e-2	-5.260e-2	-5.260e-2
$\partial I/\partial \xi_3$	-8.726e-1	-8.741e-1	-8.741e-1

Table 3: Sensitivity comparison using the coordinate strained and shock penalty objective function.

RESULTS

Design cases were run varying the number of design variables, the flow solution algorithm, the method of calculating the sensitivities, and the optimization routine. The initial values of the design variables describe the area distribution shown in figure 13. The initial distribution places the shock wave significantly far away from the target shock so that the interaction of the flow discontinuity and the discretization of the flow field on the optimization process becomes important. Figure 14, 15, and 16 show the initial velocity solution in comparison to the target for each of the flow solving algorithms. Average CPU times on a SGI Iris 340 VGX to compute the velocity field

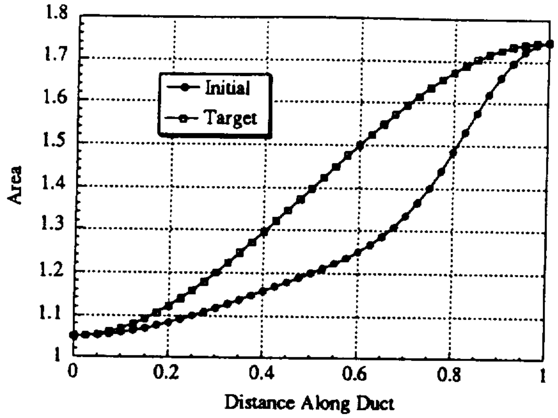

Figure 13: Initial area Distribution

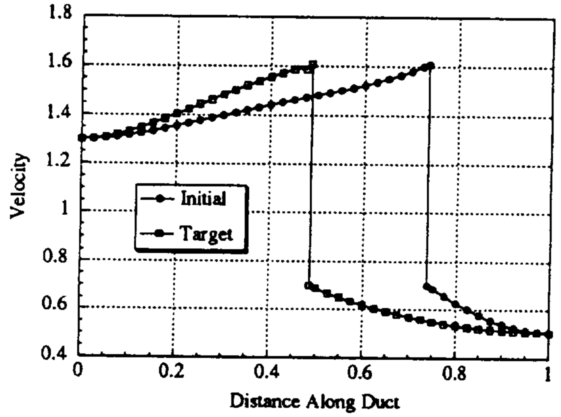

Figure 14: Initial velocity profile using exact flow solver.

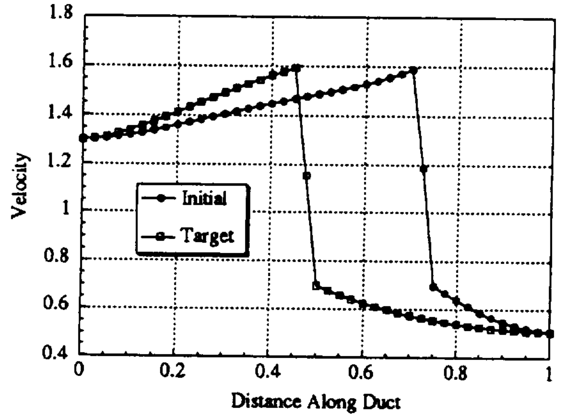

Figure 15: Initial velocity profile using Godunov flow solver.

i.e., solution to (3) are listed in table 4. The exact solution was computed in a very small amount of CPU time, less then 0.5 sec.

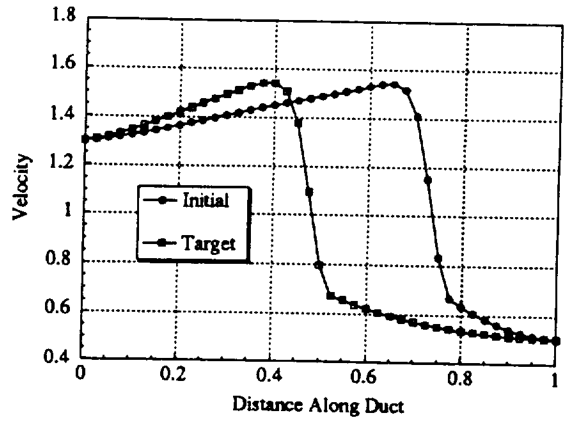

Figure 16: Initial velocity profile using artificial viscosity flow solver.

	Ave. CPU (sec.) per sol'n
Exact Solution	0
Godunov Solution	8
Artificial Viscosity Solution	3

Table 4: CPU time for flow field solutions.

Shock-Fitting

The shock-fitting objective function was tested using the finite-difference method with the SQP optimization routine. As mentioned earlier, the objective function is tailored for the case where the jump in velocity across the discontinuity is clearly defined. For this reason only exact flow solution results are presented. In comparison, the design process is attempted using the discontinuous objective function. Tables 5, and 6 list the initial, final, and targeted values of the design variables for optimization of the discontinuous and shock-fitted objective functions using 3 and 7 design variables. We see that the shock-fitted objective function will converge to the target area distribution whereas the discontinuous objective function converges to the incorrect solution. The SQP optimization algorithm allows the user to specify maximum and minimum limits on the design varaibles. In the case of the discontinuous objective funciton, many design variables were driven to these limits.

	Discontin	Shock-	
	-uous	fitted	
$\xi_{initial}$	ξ_{final}	ξ_{final}	ξ_{target}
1.1	1.1110	1.1822	1.1586
1.2	1.1824	1.4010	1.3975
1.4	1.3830	1.7662	1.6364

Table 5: Shock-fitted results for 3 design variables

	Discont.	Shock-	
		fitted	
$\xi_{initial}$	ξ_{final}	ξ_{final}	ξ_{target}
1.0643	1.0879	1.0718	1.0799
1.1000	1.1404	1.1644	1.1586
1.1471	1.3000	1.2625	1.2700
1.2000	1.2000	1.3988	1.3975
1.2658	1.2500	1.5164	1.5251
1.4000	1.4000	1.5958	1.6364
1.6202	1.8000	1.7139	1.7151

Table 6: Shock-fitted results for 7 design variables

Comparison of Sensitivity Algorithms

In the following paragraphs the design processes using the finite-difference, direct discrete, and adjoint discrete methods for computing design sensitivities are compared. Specifically, the cost measured in CPU time for computing the sensitivities and time to compute the design are used as a basis for comparison. In this comparison, the artificial viscosity algorithm is used for the computation of the flow solution and the Jacobian, J.

Table 7 lists the pertinent data for this comparison. In table 7, F-eval refers to the number of flow field solutions needed to converge the design. This number is equal to one plus the number of forward evaluations needed per iteration since an extra solution is required to check for convergence. Sensitivity CPU refers to the average time to compute one design sensitivity $\partial I/\partial \xi_j$. CPU time indicates the total time required to complete the optimization.

As expected the direct discrete method is much more efficient than the finite difference method.

CONCLUSIONS

In this paper we have developed several algorithms to calculate design sensitivities. In the application of these methods, complications involving the interaction between flow discontinuities and the discretization of the flow field cause the design process to fail. We have shown that both shock fitting and the coordinate straining and shock penalty technique can overcome these difficulties.

	No. of Design Variables	No. of Iterations	No. of F-evals	Sensitivity CPU (sec)	CPU Time
Finite-Difference	3	13	53	2.79	179
Finite-Difference	7	30	241	3.54	937
Finite-Difference	19	39	1561	3.28	2323
Direct Discrete	3	14	15	.010	97
Direct Discrete	7	32	33	.0057	229
Direct Discrete	19	39	40	.0042	243
Adjoint Discrete	3	14	15	.010	98
Adjoint Discrete	7	32	33	.0043	228
Adjoint Discrete	19	39	40	.0021	241

Table 7: Comparison of Sensitivity Algorithms using the artificial viscosity solver

Sensitivity methods are easily adaptable to coordinate straining with a shock penalty which makes the optimization algorithm efficient and robust for cases involving flow discontinuities. The finite-difference algorithm, direct discrete and adjoint discrete method provide similar design results. The finite-difference method solves the forward problem N+1 times per iteration and is impractical where the flow calculation is expensive. The direct methods, though more involved to implement, have significant time savings.

ACKNOWLEDGMENTS

We wish to thank M.D. Salas of NASA Langley Research Center for his suggestion to use shock-fitting to avoid a discontinuous objective function. This research was supported by NASA grant NAG 1-1466.

References

[1.] Frank, Paul D., & Shubin, Gregory R., "A Comparison of Optimization-based Approaches for a Model Computational Aerodynamics Design Problem," *Journal of Computational Physics*, vol. 98, 1992, pp. 74-89.

[2.] Frank, Paul D., & Shubin, Gregory R., "A Comparison of the Implicit Gradient Approach and the Variational Approach to Aerodynamic Design Optimization," Applied Mathematics and Statistics Technical Rept., AMS-TR-163, Boeing Computer Services, Seattle, WA April 1991.

[3.] Anderson, John, D., *Modern Compressible Flow with Historical Perspective*, McGraw-Hill Book Company, New York, 1982, pp. 120-150.

[4.] Nixon, D., "Perturbation of a Discontinuous Transonic Flow," *AIAA Journal*, 16, No. 1, 1978, pp. 47-52.

[5.] Stahara, S.S., "A Rapid Approximation Procedure for Nonlinear Solutions: Application to Aerodynamic Flows and Design/Optimization Problems," Chapter 18 in *Transonic Aerodynamics*, 81, Progress in Astronautics and Aeronautics, D. Nixon, Ed., AIAA, New York, 1982.

[6.] Joh, C.-Y., Grossman, B. and Haftka, R.T., "Design Optimization of Transonic Airfoils," proceedings of The International Conference on Inverse Design Concepts and Optimization in Engineering Sciences - III, G.S. Dulikravich, Ed., Washington, D.C., Oct. 1991, pp. 445-456, also to appear in *Engineering Optimization.*

[7.] Gerald & Wheatly, *Applied Numerical Analysis*, 4th ed., Addison-Wesley Publishing Company, New York, 1989, pp. 217-228.

[8.] Haftka, R. T, & Gürdal, Zafer, *Elements of Structural Optimization*, 3rd ed., Kluwer Academic Publishers, Boston, 1992.

[9.] Schittkowski, Klaus, "Nonlinear Programming Codes: Information, Tests, Performance," *Lecture Notes in Economics and Mathematical Systems*, vol. 183, Springler-Verlag, New York, 1980.